In re Application of Kelvin S. Vartti et al.
Title: System and Method for Maintaining Memory Coherency within a Multi Processor Data Processing System
File #5614 - Customer #27516
Attorney: Beth L. McMahon, Reg. 41,987, 651-635-7893
Express Mail EU039120208US
1 of 5 Drawings

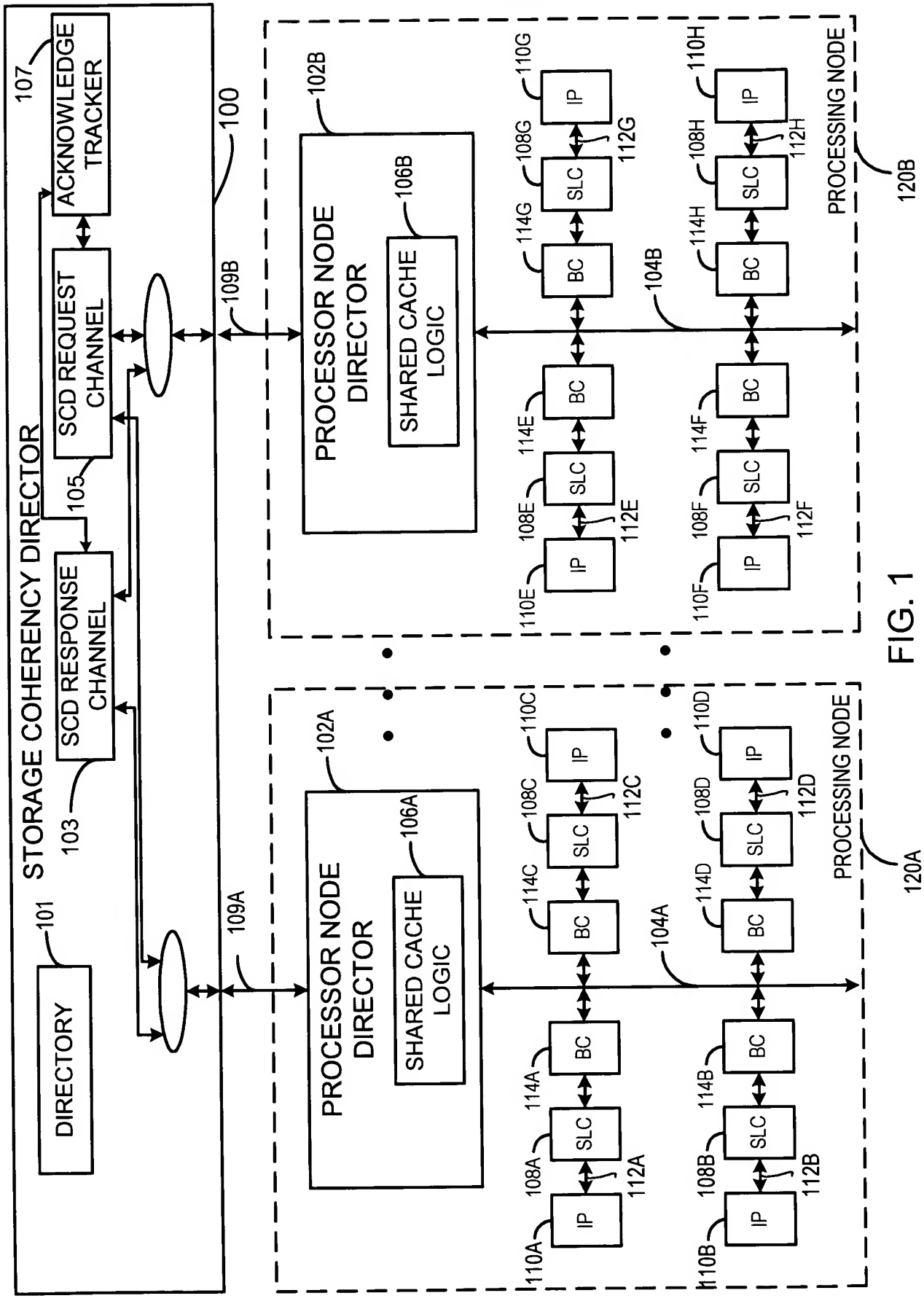

FIG. 1

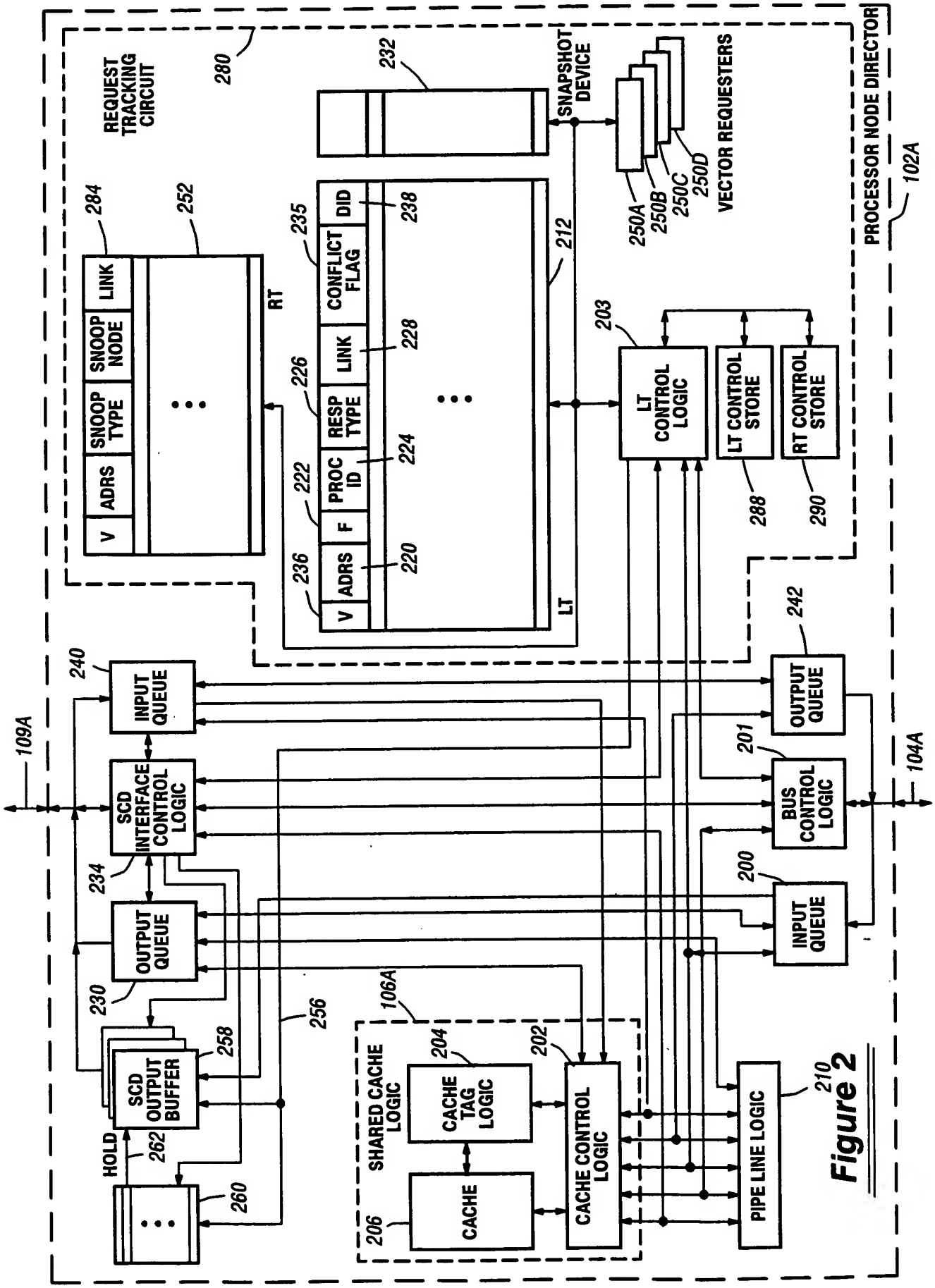
PROCESSOR NODE DIRECTOR
102A
REQUEST TRACKING CIRCUIT
280
SNAPSHOT DEVICE
232
VECTOR REQUESTERS
250A
250B
250C
250D
284
252
LINK
SNOOP NODE
SNOOP TYPE
ADRS
V
RT
235
CONFLICT FLAG
DID
238
212
226
RESP TYPE
228
LINK
222
PROC ID
224
F
220
236
ADRS
V
LT
203
LT CONTROL LOGIC
LT CONTROL STORE
RT CONTROL STORE
288
290
242
240
INPUT QUEUE
OUTPUT QUEUE
109A
104A
201
SCD INTERFACE CONTROL LOGIC
BUS CONTROL LOGIC
234
200
OUTPUT QUEUE
INPUT QUEUE
230
106A
256
SHARED CACHE LOGIC
204
202
258
SCD OUTPUT BUFFER
CACHE TAG LOGIC
CACHE CONTROL LOGIC
210
PIPE LINE LOGIC
HOLD
262
206
CACHE
260

Figure 2

In re Application of Kelvin S. Vartti et al.
Title: System and Method for Maintaining Memory Coherency within a Multi Processor Data Processing System
File #5614 - Customer #27516
Attorney: Beth L. McMahon, Reg. 41,987, 651-635-7893
Express Mail EU039120208US
3 of 5 Drawings

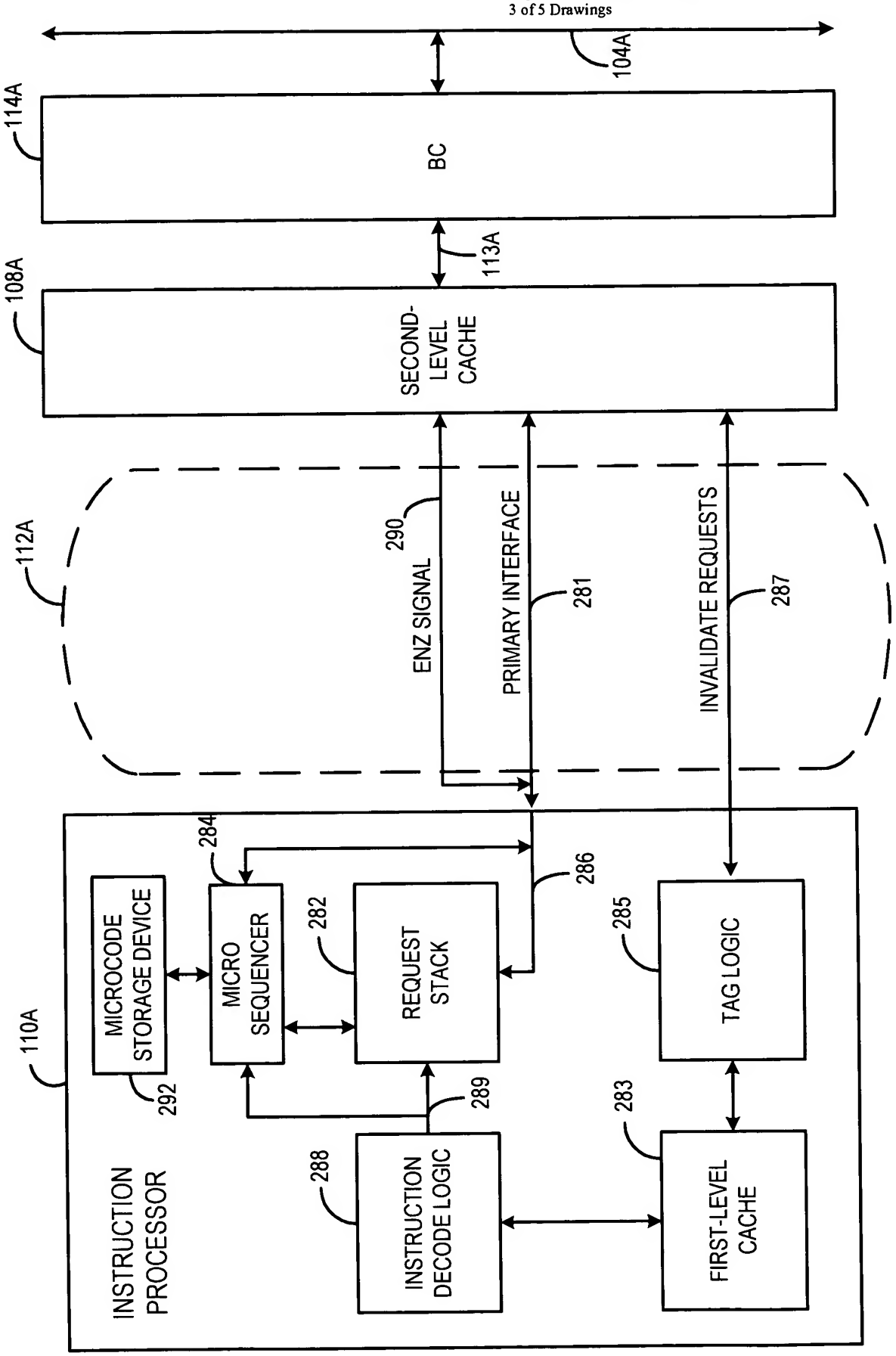

FIG. 3

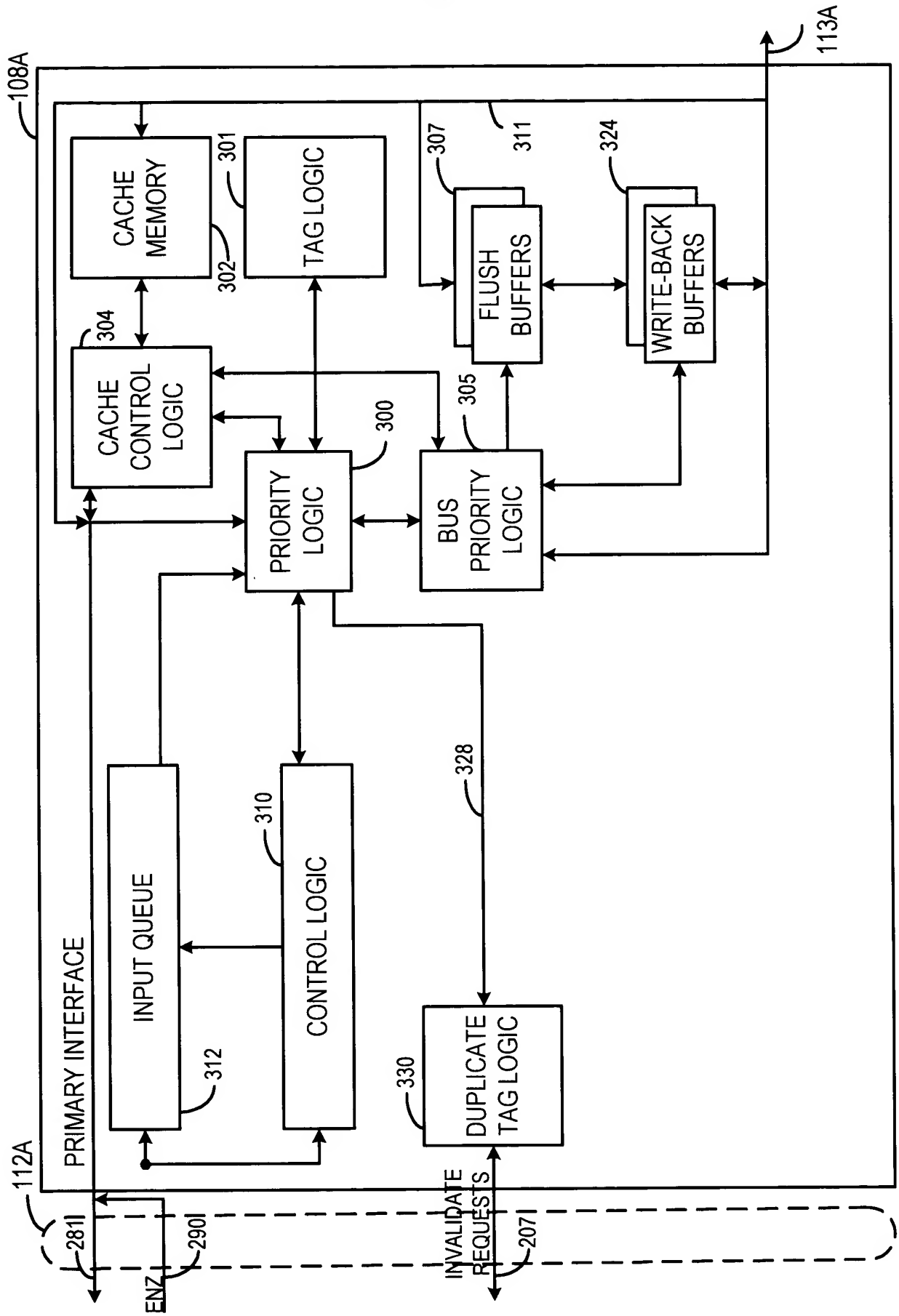
108A
113A
CACHE MEMORY
301
302
TAG LOGIC
307
311
324
FLUSH BUFFERS
WRITE-BACK BUFFERS
304
CACHE CONTROL LOGIC
300
305
PRIORITY LOGIC
BUS PRIORITY LOGIC
328
310
INPUT QUEUE
CONTROL LOGIC
312
330
DUPLICATE TAG LOGIC
112A
PRIMARY INTERFACE
281
290
ENZ
INVALIDATE REQUESTS
207

FIG. 4

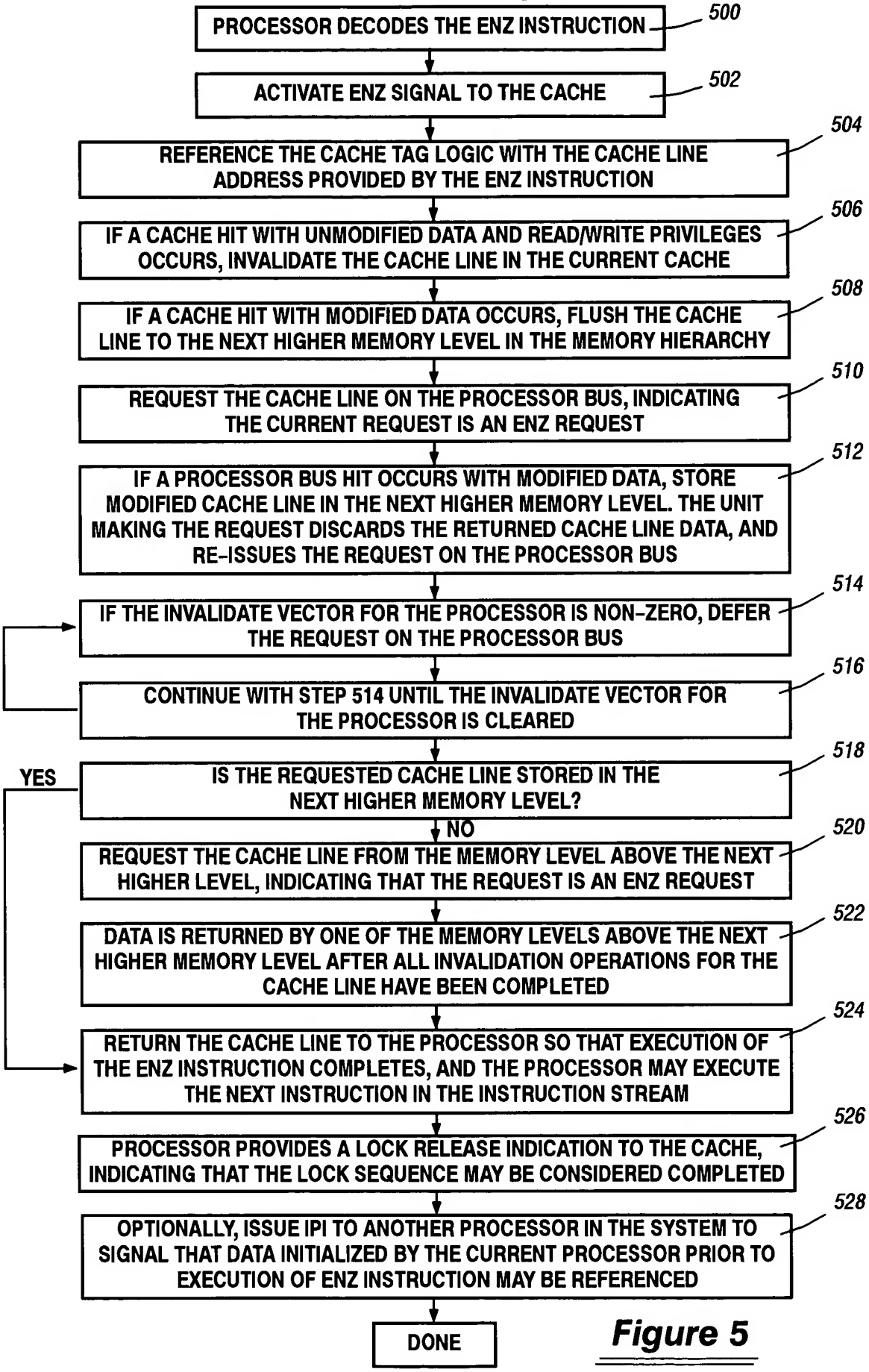
PROCESSOR DECODES THE ENZ INSTRUCTION
500
ACTIVATE ENZ SIGNAL TO THE CACHE
502
REFERENCE THE CACHE TAG LOGIC WITH THE CACHE LINE ADDRESS PROVIDED BY THE ENZ INSTRUCTION
504
IF A CACHE HIT WITH UNMODIFIED DATA AND READ/WRITE PRIVILEGES OCCURS, INVALIDATE THE CACHE LINE IN THE CURRENT CACHE
506
IF A CACHE HIT WITH MODIFIED DATA OCCURS, FLUSH THE CACHE LINE TO THE NEXT HIGHER MEMORY LEVEL IN THE MEMORY HIERARCHY
508
REQUEST THE CACHE LINE ON THE PROCESSOR BUS, INDICATING THE CURRENT REQUEST IS AN ENZ REQUEST
510
IF A PROCESSOR BUS HIT OCCURS WITH MODIFIED DATA, STORE MODIFIED CACHE LINE IN THE NEXT HIGHER MEMORY LEVEL. THE UNIT MAKING THE REQUEST DISCARDS THE RETURNED CACHE LINE DATA, AND RE-ISSUES THE REQUEST ON THE PROCESSOR BUS
512
IF THE INVALIDATE VECTOR FOR THE PROCESSOR IS NON-ZERO, DEFER THE REQUEST ON THE PROCESSOR BUS
514
CONTINUE WITH STEP 514 UNTIL THE INVALIDATE VECTOR FOR THE PROCESSOR IS CLEARED
516
IS THE REQUESTED CACHE LINE STORED IN THE NEXT HIGHER MEMORY LEVEL?
518
YES
NO
REQUEST THE CACHE LINE FROM THE MEMORY LEVEL ABOVE THE NEXT HIGHER LEVEL, INDICATING THAT THE REQUEST IS AN ENZ REQUEST
520
DATA IS RETURNED BY ONE OF THE MEMORY LEVELS ABOVE THE NEXT HIGHER MEMORY LEVEL AFTER ALL INVALIDATION OPERATIONS FOR THE CACHE LINE HAVE BEEN COMPLETED
522
RETURN THE CACHE LINE TO THE PROCESSOR SO THAT EXECUTION OF THE ENZ INSTRUCTION COMPLETES, AND THE PROCESSOR MAY EXECUTE THE NEXT INSTRUCTION IN THE INSTRUCTION STREAM
524
PROCESSOR PROVIDES A LOCK RELEASE INDICATION TO THE CACHE, INDICATING THAT THE LOCK SEQUENCE MAY BE CONSIDERED COMPLETED
526
OPTIONALLY, ISSUE IPI TO ANOTHER PROCESSOR IN THE SYSTEM TO SIGNAL THAT DATA INITIALIZED BY THE CURRENT PROCESSOR PRIOR TO EXECUTION OF ENZ INSTRUCTION MAY BE REFERENCED
528
DONE

Figure 5